VecRoad: Point-based Iterative Graph Exploration for Road Graphs Extraction

Yong-Qiang Tan   Shang-Hua Gao   Xuan-Yi Li   Ming-Ming Cheng   Bo Ren✉
TKLNDST, College of CS, Nankai University
https://mmcheng.net/vecroad/

Abstract

Extracting road graphs from aerial images automatically is more efficient and costs less than from field acquisition. This can be done by a post-processing step that vectorizes road segmentation predicted by CNN, but imperfect predictions will result in road graphs with low connectivity. On the other hand, iterative next move exploration could construct road graphs with better road connectivity, but often focuses on local information and does not provide precise alignment with the real road. To enhance the road connectivity while maintaining the precise alignment between the graph and real road, we propose a point-based iterative graph exploration scheme with segmentation-cues guidance and flexible steps. In our approach, we represent the location of the next move as a 'point' that unifies the representation of multiple constraints such as the direction and step size in each moving step. Information cues such as road segmentation and road junctions are jointly detected and utilized to guide the next move and achieve better alignment of roads. We demonstrate that our proposed method has a considerable improvement over state-of-the-art road graph extraction methods in terms of F-measure and road connectivity metrics on common datasets.

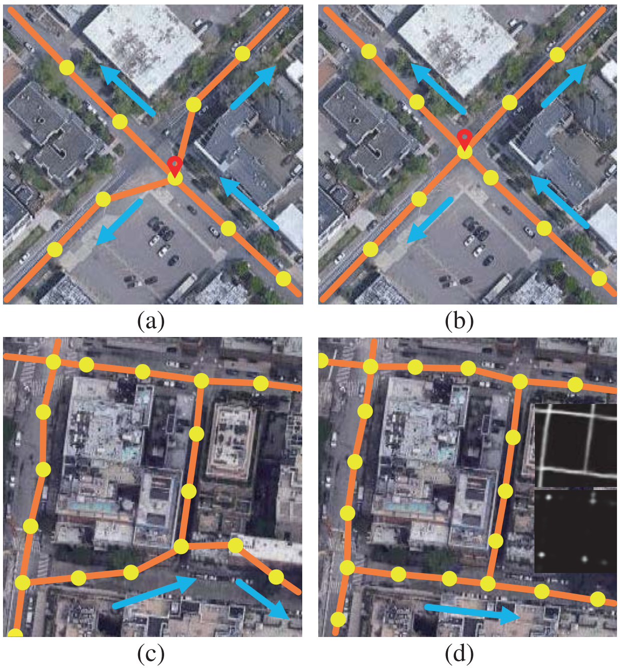


Figure 1. (a) Shifted road junctions due to the fixed moving step size. (b) Moving with our proposed flexible step size. (c) Misalignment between graphs and roads without segmentation guidance. (d) Segmentation-cues guided next move prediction generates graph with precise alignment.

1. Introduction

Road graph, the vectorized representation of road maps, allows real-world applications such as shortest-path searching for navigation. Conventionally, reliable road graphs are generated by expensive and time-consuming field acquisition and manual labeling. In recent years, convolutional neural networks (CNNs) [20, 1, 26] are adopted to automatically construct high-precision and wide-coverage road graph from aerial images with less human workload. The most common approaches [16, 2] use post-processing methods, e.g. morphological operation [31] and hard-coded rules [16, 7], to extract the road graph from skeletonized CNN-predicted road segmentation. However, the obtained graph is highly affected by the quality of segmentation, where an intermittent segmentation often leads to a graph with low connectivity. To enforce the road connectivity, recently proposed methods [1, 26] construct road graphs through iterative next move exploration. In their methods, by predicting the next move in a local patch and connecting it to the current road graph, the complete road graph is generated iteratively. However, iterative next move exploration purely focuses on local information while estimating the next move, which can result in misalignment between graphs and real roads. As revealed in Fig. 1(c), even the graph is explored in a roughly correct direction, a part of the predicted graph is outside the real road map due to their localized next move finding strategy. Also, as shown in

Fig. 1(a), the fixed step size in current methods can easily cause shifted road junctions in graphs.

To enhance the road connectivity while maintaining the precise alignment between graphs and roads, we propose a point-based iterative graph exploration scheme with segmentation-cues guidance and flexible step. We first represent the location of the next move as a 'point' that unifies the representation of multiple constraints such as direction and step size in each moving step. Our designed network learns to output a Gaussian probability distribution of multiple estimated point locations at each next move inference step. Compared with previous methods using moving angles as an indicator of next move (Fig. 2(a)), our new representation supports easy deduction to the direction and step size while avoiding complex multiple supervisions in the training. Examples of our proposed 'point' representation of next move are shown in Fig. 2(b) and (c). By supervising point coordinates, our proposed method learns to predict the correct location of the next move with a flexible step size at non-trivial points (road junctions, road ends, and linking points) to existing road graphs as shown in Fig. 1(b) and Fig. 4. Same as the inference stage, in the training phase, we can also take advantage of the global information that segmentation-cues can provide, which gives an overview of the road. Therefore, we use road segmentation and junction cues as implicit guidance to predict road graphs with accurate alignment, as revealed in Fig. 1(d). For an end-to-end design, we extract road segmentation and junction cues along with the next move predictions jointly in a unified network with a sharing backbone. Our main contributions are as follows:

- A point-based iterative next move exploration method with a flexible step size detection technique which can precisely locate on the non-trivial points during the next move exploration.
- The exploration guidance from segmentation-cues, generating road graphs with both good connectivity and good alignment precision.

2. Related Work

2.1. Road Segmentation

Extracting roads from aerial images into binary pixels is a well-studied task in the remote sensing area. Traditional methods construct road maps by various techniques such as utilizing nearby buildings and vehicles [11], shape factors [22], simulated annealing technology [23], and distinct spectral contrast and locally linear trajectory [6]. Minimum spanning tree [24], higher-order conditional random field [27, 28] and junction process [3] are also performed to construct road graphs.

Recent works apply deep learning to generate road maps with higher performance. In [17], restricted Boltzmann machine is applied for road detection, while pre-processing is adopted for dimensionality reduction of input data. Post-processing is further employed to remove disconnected blotches and fill in the holes in the roads. Saito *et al*. [20] use the CNN to directly generate road segmentation from raw remote sensing imagery without pre-processing. Cheng *et al*. [5] extract road centerline with a cascaded neural network. Zhang *et al*. [32] apply residual connections [10] to the U-Net [19] to learn more delicate features for road segmentation. The D-linknet [34] combines dilation convolutions [30] and Linknet [4] to enlarge the receptive field for road extraction from high-resolution satellite imagery.

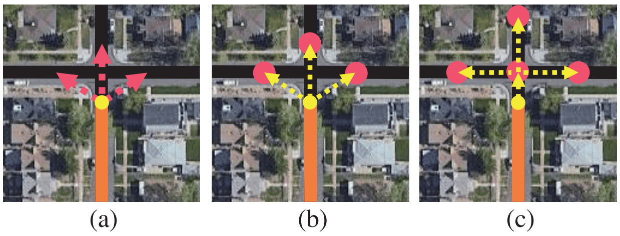
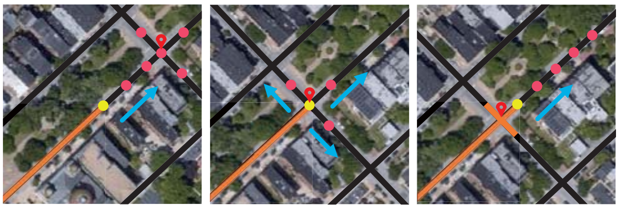


Figure 2. The representation of next move (revealed in red indicator) in exploration-based methods. (a) Angle with fixed step size; (b) Point with fixed step size; (c) Point with flexible step size. The flexible step size can better represent the road geometry and localize junctions.

2.2. Road Graph Construction

To generate a fine road graph, which is the vectorized representation of road maps, connectivity and alignment should be considered at the same time. There are two mainstream frameworks to obtain a road graph. One utilizes segmentation with post-processing, and the other transforms an aerial image to graph directly.

Post-processing from Road Segmentation. The post-processing method adopts a threshold to binarize road segmentation. Then the morphological thinning technology [31] is applied to obtain a one-pixel-wide road skeleton. To remove the redundancy of the graph, previous approaches employ the Ramer-Douglas-Peucker algorithm [7]. Máttyus *et al*. [16] use a light-weight CNN with a soft-IOU loss to generate segmentation output at the first procedure. After graph conversion, they remove short edges and reason about missing connections with A^* algorithm as the shortest path problem. Batra *et al*. [2] introduce orientation learning and erasure-refinement learning. Orientation learning endues neural networks the ability to handle the connection between pixels. Furthermore, erasure-refinement learning learns the pattern of road connection and optimizes the road segmentation output from the first step. The obtained road graphs have better connectivity on APLS metric [25].



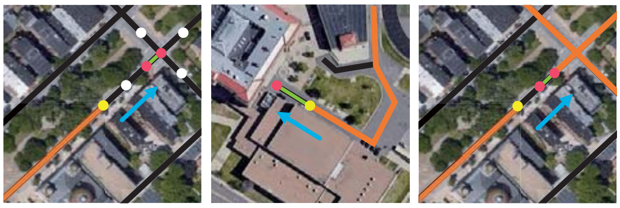
(a) ⟶ (b) ⟶ (c)

Figure 3. A continuous point-based iterative exploration from (a) to (c), the next move trajectory is revealed without connection ambiguity. Black lines are ground-truth road graph annotation, while orange lines are walked paths. Yellow point is the coordinate of current vertex. Pink points indicate junctions, and the red points are the target represented by Gaussian distribution.

Iterative Road Graph Exploration. Bastani *et al.* [1] adopt an iterative exploration algorithm to directly generate road graphs. Firstly, starting points are generated from graph ground-truth and extra road segmentation network when training and inferring, respectively. Then, they iteratively feed the cropped patch from an aerial image centered with a starting point into the neural network. One of the network outputs is a probability, deciding whether the algorithm needs to keep searching or stop. The other output is an angle vector, which represents the angles from the current vertex to next moves. They use a map-matching algorithm to keep the searching path along the current road rather than go into other roads. Li *et al.* [14] take advantage of polygons to fit the shape of roads and buildings. The polygon-based method uses a CNN-RNN architecture, to recurrently extract key points of the road geometry. Then a right-hand rule is applied to outline the road contours.

3. Method

The iterative exploration framework [1] constructs road graphs by continuously predicting the next move, and merging it into existing road graphs. We adopt this framework, and propose several schemes to improve the performance of road graph construction. We take advantage of the point-based next move representation, which is a unified combination of moving angle and distance. Owing to the point representation, multiple constraints can be easily applied without complex supervision. We propose a flexible step size detection technique which is designed to dynamically align with the junction at the training phase. We utilize road and junction segmentation cues to guide the exploration and achieve better alignment of roads. We unify those schemes into our proposed framework, Road Point Network (RPNet), to generate road graphs with high connectivity and precise alignment. In this section, we will first recap the iterative exploration framework, then describe the details of our method.



(a) (b) (c)

Figure 4. The flexible step size (illustrated with green lines) is adopted when encountered (a) road junctions, (b) road ends, and (c) linking points in the next move. The orange lines are walked paths, which is important prior to the connection of linking points.

3.1. Overview of Iterative Exploration

Road graph G is a vectorized representation of road maps, which contains a vertex set $V = \{v_1, v_2, \cdots, v_n\}$, and an edge set $E = \{e_1, e_2, \cdots, e_m\}$. An edge e is a straight-line between two vertices, denoting the road between those two vertices. The road graph is constructed through iteratively exploring new vertex along the road and add the new vertex to the existing road graph G with an edge between two vertices. Specifically, the iterative exploration starts with a starting vertex set S indicating starting points of exploration. Commonly, S is obtained from peak points of road segmentation [1] or junction segmentation. The vertex set V in G is initiated as a copy of S. For each exploration, a vertex v is popped from S as the starting point. A neural network takes an aerial image patch centered with this vertex as the input, and predict the next vertex set V'. If the predicted $v \in V'$ has a matched vertex in the same region in V, the matched vertex will be adopted as the newly obtained vertex. Then, the newly obtained vertex and the road between the existing and new vertex are added to V and E respectively to form the new G. S is updated by $S \cup V'$. A new starting vertex is obtained from S to start a new exploration. The exploration ends when S is empty. In Fig. 3, we present the dynamic process in the exploration. With the exploration moving on, the graph is constructed iteratively.

3.2. Point-based Iterative Exploration

Point-based Next Move Prediction. In this work, we represent the location of the next move as a ‘point’ that unifies the representation of both moving angle and distance, as shown in Fig. 2(b). In the training phase, the supervision of the next move is always set in the centerline of roads, so the output is assured to trace the real road iteratively. The exploration detector is trained with the supervision of Gaussian distribution centered with the position of the next move. Taking the point-based exploration as a pixel-wise task, the neural network can precisely predict an “in-road” next move. During inference, the position of the next move can be obtained from the peak of the predicted distribution.

It is easy to apply multiple constraints (e.g., direction and step size) on point representation at the training phase without complex multiple forms of supervision. In the following, we will discuss our method in detail.

Flexible Step Size Scheme. In [1], an angle classifier with a fixed step size is applied to detect up to 64 vertices as shown in Fig. 2(a). There are several kinds of non-trivial points in roads such as junctions, road ends and linking points as revealed in Fig. 4. The road length between the current position and nearby non-trivial next move can hardly match the integral multiple of the fixed step size. As an example shown in Fig. 1(a), detector with a fixed step size may generate misaligned graphs with real roads whenever meeting a junction in the next move. To ensure the precise alignment of roads and junctions, we design such a flexible step size scheme. During the training phase, we perform the exploration on an empty graph with the supervision of a ground-truth graph. In every exploration step, we dynamically follow the ground-truth graph to generate the next move supervision. We denote the fixed step size as s here, and the flexible step size is designed to be an adjustable length between $0.5 \times s$ and $1.5 \times s$. When there is a non-trivial point within 1.5 scale of s from the current vertex, we generate the supervision Gaussian distribution exactly on the detected point. With a flexible step size, non-trivial points such as junctions can be easily handled and thus the graph will align with the real road. This scheme also helps enhance graph connectivity. One specific case is shown in Fig. 4(c), with the flexible step size, a broken endpoint of a previous interrupted exploration can be easily matched and connected. As to the case when there are no non-trivial point in the next exploration area, in Fig. 3(c), we generate new supervision Gaussian distribution along the ground truth graph using the fixed step size from the starting point. As a conclusion, we use fixed step supervision in the middle of the road and switch to flexible step size near non-trivial points.

An extra step size learning must be carefully designed if conventional approaches that adopt angle learning want to obtain such a flexible step. On the contrary, instead of adding a moving distance detection branch, owint to our proposed 'point' representation, the point-based detector can learn a well-performing flexible step size through training with the point-based supervision encoded with moving distance as shown in Fig. 2(c).

Trajectory Exploration. In the framework of iterative exploration, every single step may bring a slight error. Inspired by the long-range reward and experience replay mechanism in reinforcement learning [18], we propose to predict a trajectory of moves at once instead of only one step. We realize this by recurrently send back the down-sampled next move prediction to the next move detector (an hourglass block) up to T times. It should be noted that, given an aerial image as input, we only extract image features once. By using the recurrent mechanism, the neural network will obtain a longer sight of future trajectory and reduce the total error.

3.3. Segmentation Cues

Different from the exploration mechanism which focuses on the local next move (in Fig. 2), the segmentation technique enjoys a more global view of interests. Comparing result of [1] in Fig. 1(c) and our result (d), we can see that without a global knowledge about where to explore in a long-term view as to be described below, the exploration will cause misalignment on both road and junction.

Road Segmentation Cue. The target of road segmentation is to extract the centerline of the road [20, 5] from aerial images. As shown in Fig. 1(b), the road centerline can better represent the topology of the road graph on a macroscopic point of view. Here, we explain two key insights on using road segmentation in our method. First of all, the iterative exploration methods mainly care about the position of the local next move, but lacks an overall knowledge of road areas, i.e. where the real roads lie. Specifically, in Fig. 7(e), (g) and (h), the global guidance of road area from road segmentation can reduce the misalignment with the real road. Secondly, in the form of a next move prediction, the road segmentation can be viewed as an ideal-choice set of exploration points. Thus a road segmentation can provide proper guidance and a centerline prior to exploration.

Junction Segmentation Cue. The junction segmentation, which is just suitable for our flexible step method, can guide the prediction of the next move when a junction is ahead. Since road junctions in the aerial images are often in the form of an area, a junction segmentation cue can help the network precisely learn the best junction location during training. For example, without the help of junction cues, when several road segments meet at the junction area, methods of exploration may detect less or more crosses. As shown in Fig. 7(e), (f) and (h), without the junction segmentation as support, the exploring schedule struggles at a complex junction, and with the guidance of junction, the predicted graph is more methodical. Similar to the reasons for using road segmentation, junction segmentation can give a prior of the junction location, which helps the neural network better recognize the distance pattern and decide the step size to reach the accurate coordinate of junctions.

3.4. Network Architecture

In our network architecture design, as shown in Fig. 5, a side fusion $\mathcal{F}$ of VGG backbone is adopted to extract

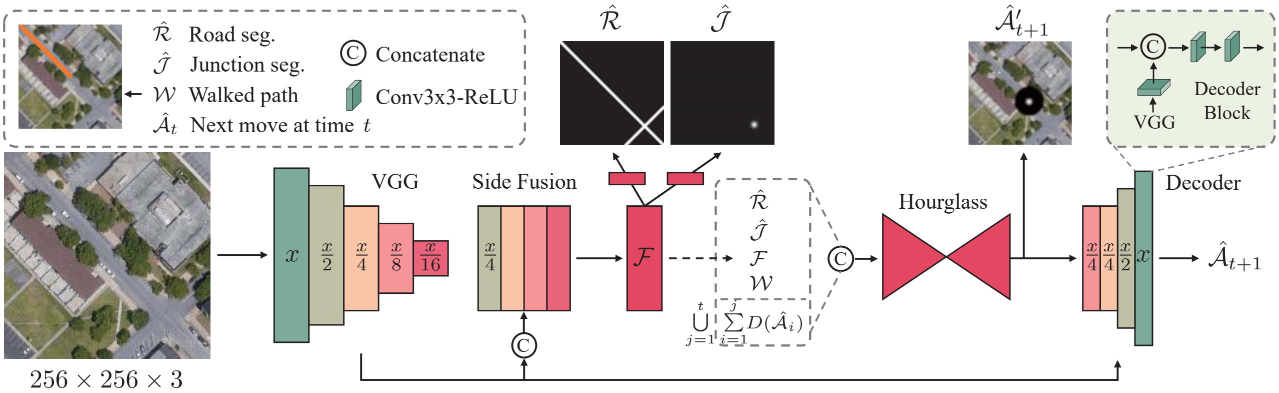

$\hat{\mathcal{R}}$ Road seg.
$\hat{\mathcal{J}}$ Junction seg.
$\mathcal{W}$ Walked path
$\hat{\mathcal{A}}_t$ Next move at time t
© Concatenate
Conv3x3-ReLU
$256 \times 256 \times 3$
VGG
Side Fusion
x
$\frac{x}{2}$
$\frac{x}{4}$
$\frac{x}{8}$
$\frac{x}{16}$
$\mathcal{F}$
$\hat{\mathcal{R}}$
$\hat{\mathcal{J}}$
$\mathcal{W}$
$\bigcup_{j=1}^{t}\sum_{i=1}^{j} D(\hat{\mathcal{A}}_i)$
Hourglass
$\hat{\mathcal{A}}'_{t+1}$
Decoder
Decoder Block
VGG
$\hat{\mathcal{A}}_{t+1}$

Figure 5. Overview of RP-Net architecture. The rectangles are feature maps of different scales. The color of each feature map means the usage of corresponding VGG stages. The walked path is a binary mask but visualized as an orange line on the input image. The recurrent part takes $\bigcup(\mathcal{F}, \hat{\mathcal{R}}, \hat{\mathcal{J}}, \mathcal{W}, \bigcup_{j=1}^{t}(\sum_{i=1}^{j} D(\hat{\mathcal{A}}_i)), \bigcup_{i=1}^{T-t}(\hat{\mathcal{A}}_0))$ as input, and outputs predicted next moves $\hat{\mathcal{A}}_{t+1}$ by time series.

the pyramid basic feature of the aerial image at the quarter scale. To make the neural network predict a more robust forward direction, we explicitly generate the explored path segmentation $\mathcal{W}$ from the inferred graph as an intermediate input. we utilize the side output features fusion $\mathcal{F}$ to produce a quarter-scale road and junction segmentation prediction, which are denoted by $\hat{\mathcal{R}}$ and $\hat{\mathcal{J}}$, respectively. We adopt road supervision $\mathcal{R}$ and junction supervision $\mathcal{J}$ to guide the backbone network to learn the basic representation of a road. The hourglass block is a pyramid feature auto-encoder, which basically takes the concatenation of road backbone feature and $\mathcal{W}$ as input, fuse features through a quick down-sampling and up-sampling. When adopting segmentation cues as implicit guidance, the input of our hourglass block is further concatenated by segmentation cue feature maps as $\bigcup(\mathcal{F}, \hat{\mathcal{R}}, \hat{\mathcal{J}}, \mathcal{W})$. Here, $\bigcup(\cdot)$ denotes concatenation at channel level. Here we do not distinguish intermediate segmentation feature maps and output prediction in the representation $\hat{\mathcal{R}}$ and $\hat{\mathcal{J}}$ for simplicity. The implicit guidance of $\hat{\mathcal{R}}$ and $\hat{\mathcal{J}}$ will act as the input and brings good interpretability to this design. After the hourglass block, a rough Gaussian distribution of next move $\hat{\mathcal{A}}'$ will be guaranteed by the side supervision of a real Gaussian map $\mathcal{A}$. The decoder part of our network is designed to magnify and refine the prediction with both high-level and low-level road information, helping the predicted distribution to be generated precisely. Finally the supervision $\mathcal{A}$ will be applied to guarantee a meticulous distribution $\hat{\mathcal{A}}$. Moreover, owing to the joint learning of multi-task, our method is trained end-to-end without a separate network for the obtaining of the starting point.

To recurrently predict T steps, we down-sample the final prediction as a quarter scale and reuse it through the concatenation with the input of hourglass block mentioned above. We use a placeholder $\hat{\mathcal{A}}_0$ initially to ensure the consistency of feature channel. Given $\bigcup(\mathcal{F}, \hat{\mathcal{R}}, \hat{\mathcal{J}}, \mathcal{W}, \bigcup_{j=1}^{t}(\sum_{i=1}^{j} D(\hat{\mathcal{A}}_i)), \bigcup_{i=1}^{T-t}(\hat{\mathcal{A}}_0))$, we obtain a next moves probability map $\hat{\mathcal{A}}_{t+1}$. Here, $D(\cdot)$ denotes the down-sampling operation. Therefore we can recurrently get $\hat{\mathcal{A}}_t$ of T time steps. If the road segment meets a junction before T time steps, say, at step $t = k$, $k < T$, we will ignore the steps $t > k + 1$ when calculating the loss function because supervision after the exploration $t = k+1$ will be ambiguous for vertex connection.

As to the detail of the network architecture design, the hourglass block is constructed by 4-layer down-sampling and 4-layer up-sampling with the residual connection. Each layer contains two Conv-ReLU layers with a kernel size of 3. Every decoder block takes 32-channels backbone features and 32-channels next move features calculated from the previous block as a sum, and is followed by two 3×3 convolution layers. We use standard binary cross-entropy loss to optimize the $\hat{\mathcal{R}}$, $\hat{\mathcal{J}}$, and $\sum_{t=1}^{T} \hat{\mathcal{A}}_t$ respectively. The total loss function is

$$\mathcal{L} = \sum_{t=1}^{\mathcal{T}} \Big(L(\hat{\mathcal{A}}_t, \mathcal{A}_t) + L(U(\mathcal{A}'_t), \mathcal{A}_t) \Big) + \lambda_1 L(\hat{\mathcal{R}}, \mathcal{R}) + \lambda_2 L(\hat{\mathcal{J}}, \mathcal{J}), \tag{1}$$

where $L(X, Y)$ is the binary cross-entropy loss between prediction matrix X and ground-truth matrix Y. $U(\cdot)$ denotes the up-sample function and λ is a parameter to balance multi-class loss. In Equ. (1), $\mathcal{T}$ is determined by $min(k + 1, T)$. The λ_1 and λ_2 in loss function are both set to 1.

3.5. Implementation Details

Following RoadTracer [1], we dynamically generate ground-truth next move to train the neural network. Here, during training, we adopt an empty graph to explore and a supervision graph to guide the coordinate of the next move. To ensure the independently identically distribution of training process, we sample the batch of training patches from different aerial images and apply random pop from the starting point set of each training image. Since a random graph exploration is naturally a data augmentation technology, we apply no extra data augmentation.

The RP-Net is trained with cropped patch from aerial images of 256 × 256 resolution, which is a trade-off between effectiveness and efficiency. We adopt Pytorch framework and the released VGG-16 model [21] as an initialization. We train the network with Adam optimizer [13] for 102,400 iterations. We start training with the initial learning rate of $1e-3$, and we drop the learning rate one time by the factor of 0.1 at the 40,960 iteration. We use a batch size of 24 to train the model with 2 NVidia Titan XP. There is no data augmentation applied to the training data. We employ a threshold 0.4 for the transformation from the next move probability map to coordinates when inferring. The value of the total time step T is a trade-off between image size and step size. We use $T = 4$ in this work to provide enough trajectory length and make sure the next move is within the input image at the same time. It should be noted that a larger image size requires more GPU resources while maintaining the batch size. In the training phase, the fixed step size has a distance of 20 pixels, and the flexible step length is adjusted between 10 and 30 pixels to dynamically fit the distance from the end of current road segment. Thanks to our segmentation cues, we do not need an extra network to extract the starting points because we can obtain S from the peak side-output of junction and road segmentation. The parameter number in our network is 20M, less than 21M (for starting points generation) plus 26M (for iterative exploration) in [1].

4. Experiments

We quantitatively and qualitatively verify our method on the RoadTracer dataset [1] and give details in this section.

4.1. Evaluation Metrics

We evaluate our results on both road alignment and graph connectivity. Following [17, 20], we adopt the pixel-metric to study the pixel-level alignment of road centerline mask. Here, the road centerline mask is generated through drawing the road graph on a 2D map with a fixed width of 8 pixel. The road width is viewed a relaxation of road centerline.

To better evaluate the road connectivity and topology, we follow [1] to evaluate the junction-level precision-recall. The $F_{correct}$ and F_{error} in [1] can be further utilized to calculate the F-score by viewing $1 - F_{error}$ as precision and $F_{correct}$ as recall. We uniformly adopt mean F-score to represent the comprehensive performance of both precision and recall on pixel-metric as '**P-F1**' and junction-metric as '**J-F1**'. More details about the pixel-metric and junction-metric will be explained in the supplementary materials.

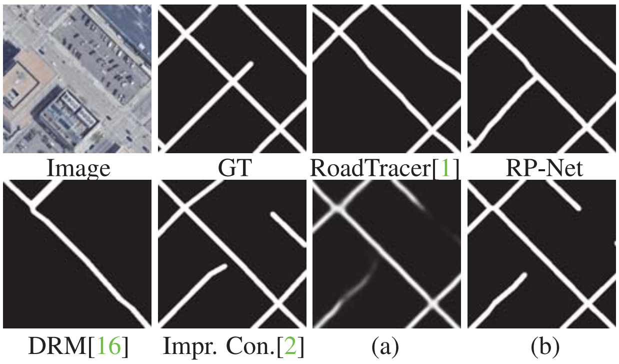


Figure 6. Qualitative comparison of various methods mentioned in Tab. 1. (a) Segmentation result generated from RP-Net-Seg., (b) graph generated from RP-Net-Seg. using post-possessing.

The Average Path Length Similarity metric (**APLS**) is introduced from [25]. Having all pairs of corresponding nodes from predicted graph $\hat{G}$ and ground-truth graph G respectively, the APLS metric studies the shortest path length difference between them:

$$APLS = \frac{1}{N}\sum\left(\frac{2}{\frac{1}{S_{\hat{G}\to G}}+\frac{1}{S_{G\to\hat{G}}}}\right), \qquad (2)$$

where

$$S_{\hat{G}\to G} = 1 - \frac{1}{M}\sum \min\left(1, \frac{|L(a,b)-L(\hat{a},\hat{b})|}{L(a,b)}\right) \qquad (3)$$

is a shortest path length score mapping from $\hat{G}$ to G. In Equ. (3), M is the number of unique paths in the mapped graph $\hat{G} \to G$. $L(\hat{a},\hat{b})$ and $L(a,b)$ means the length of the path $(\hat{a},\hat{b})$ in $\hat{G}$ and (a,b) in G, respectively. In Equ. (2), N is the number of images belonging to the dataset.

4.2. Comparison with State-of-the-art Techniques

We compare our approach to previous state-of-the-art techniques [16, 1, 2]. As shown Tab. 1, we observe that our method outperforms the state-of-the-art techniques on all the three evaluation metrics. Especially, our method has superior performance on the junction-metric, showing advantage of the point-based method in recovering road junction features.

For the methods used in the comparison, we present a quantitative visualization in Fig. 6. More visualization comparision will be given in the supplementary materials. DeepRoadMapper[16] is a segmentation-based method. Owing to the lack of connectivity information in segmentation supervision, the neural network has intermittent outputs when shadow or occlusion occurs. RoadTracer[1] is a more robust method to extract graphs from aerial images, but the approach did not consider the shift of junctions owing to the

Method	P-F1	J-F1	APLS
DeepRoadMapper [16]†	56.85	29.05	21.27
RoadTracer [1]	55.81	49.57	45.09
RoadTracer-256 [1]	59.69	52.19	-
ImprovedConnectivity [2]†	73.35	55.21	56.89
RP-Net-Seg.†	71.61	50.16	49.68
RP-Net-Full	**73.69**	62.36	61.14
RP-Net-Full+Res2Net	72.56	**63.13**	**64.59**

Table 1. Performance comparison on the RoadTracer road dataset. 'P-F1' and 'J-F1' denote the F-score of road pixel-level metric and junction-level metric, respectively. † means using the post-processing implementation from the [1]. RP-Net-Full+Res2Net means additional use Res2Net [8] as backbone feature extractor.

fixed step size of the design of angle-learning. Note that the result of RoadTracer narrows the input scale, so we fairly evaluate its performance by also narrowing the ground-truth and report as "RoadTracer-256" in Tab. 1. The [2] method is a segmentation-based method, which also adopts complex hard-coded post-processing. Although road segmentation could take connectivity into consideration and generate a well-performed road mask, the complex post-processing causes geometry deformation as well. The version with Res2Net[8] backbone sacrifices high resolution in pixel-level score slightly but achieves much better connectivity score because of multi-scale aggregation and adaptive receptive fields [33].

4.3. Ablation Study

We study the improvement of our methods through an incremental application of them. As shown in Tab. 2, initially, we perform a baseline experiment without flexible step, segmentation cue and trajectory exploration.

Flexible Step Size. By taking advantage of the flexible step size on the baseline method, the pixel-wise road alignment and junction-wise connectivity are both improved. Here, the junction-wise connectivity can be revealed from the 'J-F1' score. Specifically, the junction-metric improves 7.58% and mainly benefits from the precise junction location. As is demonstrated Fig. 8, junction shifting is significantly restrained.

Segmentation Cues. We study the effectiveness of road segmentation and junction segmentation through performing the techniques based on the flexible step method. We first only apply the road segmentation cue, i.e. apply a side supervision on the fused VGG feature and the concatenation of the road segmentation feature as the input of the next move predictor as described in Sec. 3.3. Using road segmentation cue alone, the pixel-metric improves 11.86% and the junction-metric improves 12.38%. Secondly, we only apply the junction segmentation cue, which helps precisely positioning junction and further help the flexible step method to find an accurate step size. Both metrics similarly improves but the pixel-metric improves less then that of using road segmentation cue alone. This is expectable because false junction location always results in false road alignment, but the reverse is not true. So enforcing road alignment using road segmentation has a larger effect.

Flexible Step	Road Seg.	Junc Seg.	Traj. Exp.	P-F1	J-F1	APLS
				53.28	36.25	34.69
✓				56.42	43.83	46.22
✓	✓			68.28	56.21	49.46
✓		✓		61.28	55.49	50.75
✓	✓	✓		69.81	59.42	57.28
✓	✓	✓	✓	**73.69**	**62.36**	**61.14**

Table 2. The incremental improvement of our proposed methods. Note that all the experiment share the same starting point set obtained from the full model.

However, only by combining the two segmentation cues can we observe a large increase in the APLS metric. The reason behind this is that the two segmentation cues have complementary emphasises. The road segmentation cue guided improvement mainly focuses on the alignment of road centerline, helping to avoid aborting due to out-of-road start points. On the other hand, the junction segmentation focuses on providing unified and accurate junctions from complex junction areas in the input image, which helps the existing road graph correctly links to a new junction. They both help enhance connectivity in the road graph.

The experiment result shows that all the three evaluation metrics have a performance improvement. Our point-based iterative exploration scheme naturally allows us to take advantage of the segmentation cues in a unified and compatible manner. It also endows the neural network output good alignment and connectivity properties.

Trajectory Exploration. We compare our trajectory exploration scheme to the model that employs flexible step and both segmentation cues, but has no recurrent mechanism and directly output up to T channels of "sequentially step-forward" point estimations.

We set $T = 1$ in the straight-forward model and record its junction-metric score. Then we set $T = 4$ in the training stage of the straight-forward model and evaluate the junction score four times by separately using the first 1-4 channels of output to construct the final graph. We also evaluate the junction score four times on our RP-Net by only using the first 1-4 times of recursion outputs. The results are shown in Tab. 3, which suggest that the recurrent scheme helps further improve the performance of our network.

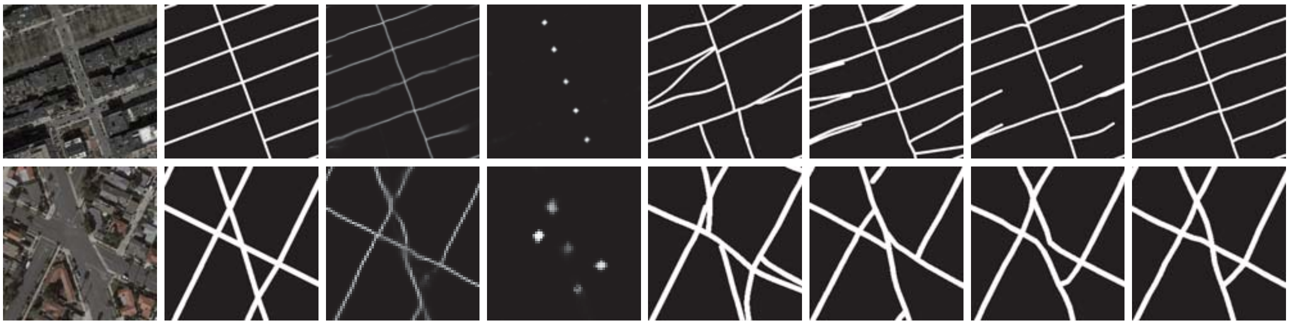


Figure 7. Visualization of road graphs with segmentation-cues guidance. We denote F as applying flexible step, R as applying road segmentation cue, and J as applying junction segmentation cue. (a) aerial image, (b) ground-truth segmentation mask, (c) predicted road segmentation, (d) predicted junction segmentation, (e) graph with F, (f) graph with F+J, (g) graph with F+R, (h) graph with F+R+J. Note that the sampled patches may be cropped from different resolution.

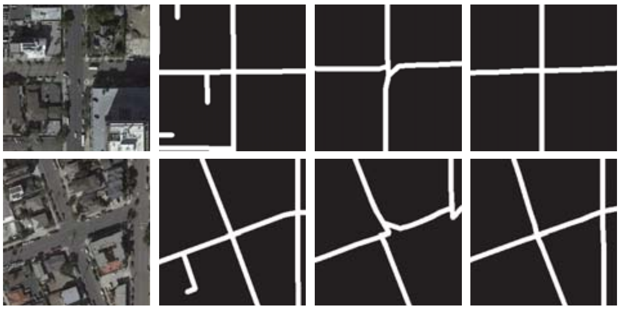


Figure 8. Visualization of graphs generated with fixed step and flexible step. (a) aerial image, (b) ground-truth segmentation mask, (c) predicted graph with fixed step size, (d) predicted graph with flexible step size.

Post-processing v.s. Full RP-Net Scheme. Note that our network also allows recovering road segmentation through performing the segmentation supervision at the decoder without other supervision. The segmentation output with the original size of aerial image can be followed by conventional post-processing techniques to generate a road graph. We evaluate the performance of this post-processing scheme and compare it to our full RP-Net scheme.

Same as other segmentation-based methods [16, 15], we generate road segmentation, with a morphological thinning and RDP algorithm to obtain graphs. After that, techniques like short edge pruning and small hole elimination [16] are applied. The metric scores are recorded in Line “RP-Net-Seg.” in Tab. 1 , which are lower than using the iterative RP-Net especially on the junction-metric and the APLS metric. A qualitative comparison is also given in Fig. 6 (a) and (b), where vague segmentation results in interrupted road graph. Using our designed full network structure, the RP-Net scheme show better connectivity and alignment.

Traj. Exp.	Train T	Test T	J-F1
×	1	1	59.19
×	4	1	60.71
×	4	2	60.69
×	4	3	60.70
×	4	4	60.44
✓	4	1	60.66
✓	4	2	61.60
✓	4	3	61.92
✓	4	4	**62.36**

Table 3. Trajectory exploration ablation study. Train T means the number of supervision channels, and Test T indicates the only utilization of first T channels of output for graph construction when inferring. We evaluate the performance of different numbers of steps adopted in trajectory exploration on the ‘J-F1’ score.

5. Conclusion

In this paper, we propose a point-based iterative aerial image exploration featuring usage of flexible step and segmentation cues. Experiments on various metrics show our approach provide significant improvement on the road graph alignment and connectivity compared to state-of-the-art methods on the RoadTracer dataset. In the future, we plan to further study the trajectory exploration and investigate the possibility of global [29] optimization in the exploration. Our high quality road detection could also be serve as a strong prior knowledge for detecting distict target in aerial images [9, 12].

Acknowledgements

This research was supported by Major Project for New Generation of AI under Grant No. 2018AAA0100400, NSFC (61922046), and Tianjin Natural Science Foundation (17JCJQJC43700).

References

[1] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *CVPR*, 2018. 1, 3, 4, 5, 6, 7

[2] A. Batra, S. Singh, G. Pang, S. Basu, C. Jawahar, and M. Paluri. Improved road connectivity by joint learning of orientation and segmentation. In *CVPR*, pages 10385–10393, 2019. 1, 2, 6, 7

[3] D. Chai, W. Forstner, and F. Lafarge. Recovering line-networks in images by junction-point processes. In *CVPR*, pages 1894–1901, 2013. 2

[4] A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *IEEE Visual Commun. Image Process.*, 2017. 2

[5] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Trans. Geosci. Rem. S.*, 55(6):3322–3337, 2017. 2, 4

[6] S. Das, T. Mirnalinee, and K. Varghese. Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images. *IEEE Trans. Geosci. Rem. S.*, 49(10):3906–3931, 2011. 2

[7] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: Int. J. Geog. Inform. Geovisualization*, 10(2):112–122, 1973. 1, 2

[8] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, 2020. 7

[9] L. Han, P. Tao, and R. R. Martin. Livestock detection in aerial images using a fully convolutional network. *Computational Visual Media*, 5(2):221–228, 2019. 8

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[11] S. Hinz and A. Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS J. Photogramm. Rem. S.*, 58(1-2):83–98, 2003. 2

[12] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019. 8

[13] D. Kinga and J. B. Adam. A method for stochastic optimization. In *ICLR*, volume 5, 2015. 6

[14] Z. Li, J. D. Wegner, and A. Lucchi. Topological map extraction from overhead images. In *ICCV*, 2019. 3

[15] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE TPAMI*, 41(8):1939–1946, 2019. 8

[16] G. Máttyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *ICCV*, 2017. 1, 2, 6, 7, 8

[17] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *ECCV*, pages 210–223. Springer, 2010. 2, 6

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 4

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 2

[20] S. Saito, T. Yamashita, and Y. Aoki. Multiple object extraction from aerial imagery with convolutional neural networks. *Electronic Imaging*, 2016(10):1–9, 2016. 1, 2, 4, 6

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[22] M. Song and D. Civco. Road extraction using svm and image segmentation. *Photogramm. Eng. Rem. S.*, 70(12):1365–1371, 2004. 2

[23] R. Stoica, X. Descombes, and J. Zerubia. A gibbs point process for road extraction from remotely sensed images. *IJCV*, 57(2):121–136, 2004. 2

[24] E. Türetken, F. Benmansour, and P. Fua. Automated reconstruction of tree structures using path classifiers and mixed integer programming. In *CVPR*, pages 566–573. IEEE, 2012. 2

[25] A. Van Etten, D. Lindenbaum, and T. M. Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018. 2, 6

[26] C. Ventura, J. Pont-Tuset, S. Caelles, K.-K. Maninis, and L. Van Gool. Iterative deep learning for road topology extraction. *BMVC*, 2018. 1

[27] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. A higher-order crf model for road network extraction. In *CVPR*, pages 1698–1705, 2013. 2

[28] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. Road networks as collections of minimum cost paths. *ISPRS J. Photogramm. Rem. S.*, 108:128–137, 2015. 2

[29] H. Wu, X. Lyu, and Z. Wen. Automatic texture exemplar extraction based on global and local textureness measures. *Computational Visual Media*, 4(2):173–184, 2018. 8

[30] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[31] T. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, 1984. 1, 2

[32] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geosci. Rem. S. L.*, 2018. 2

[33] W. Zhen, S. Yao, and J. Lin. Learning adaptive receptive fields for deep image parsing networks. *Computational Visual Media*, 4(3):1–14, 2018. 7

[34] L. Zhou, C. Zhang, and M. Wu. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *CVPR*, pages 182–186, 2018. 2